

# A05 Metaheurísticas: Multi-start e GRASP

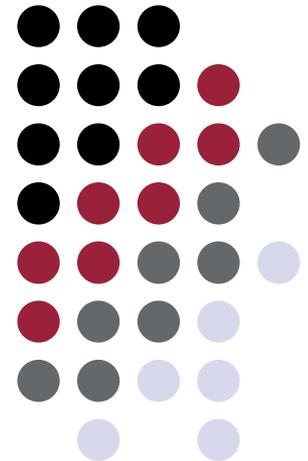


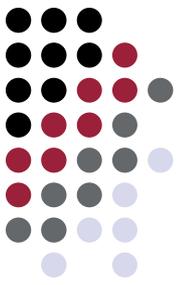
UFOP

Universidade Federal  
de Ouro Preto

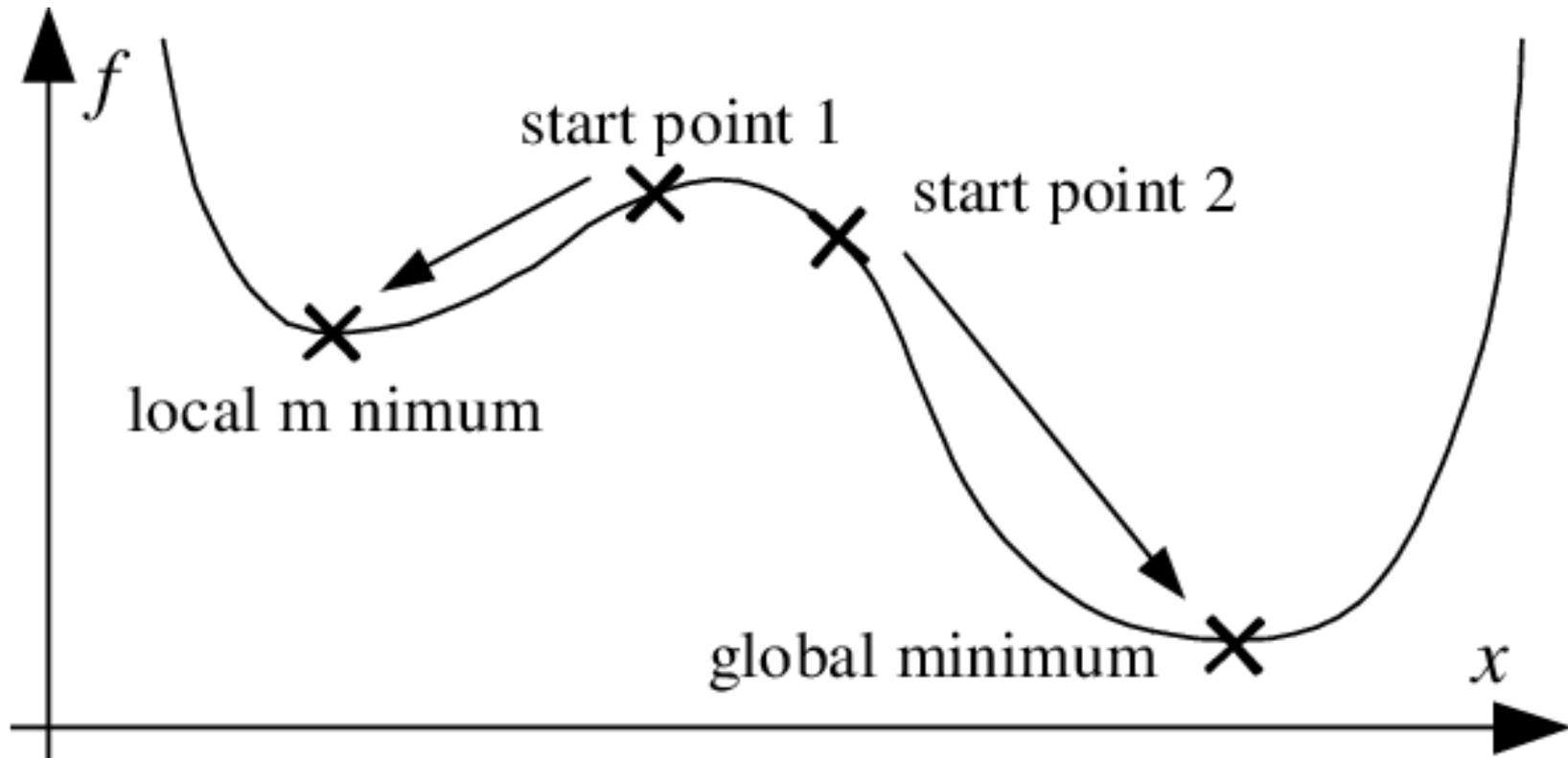
## PEP300 Técnicas Metaheurísticas para Otimização Combinatória

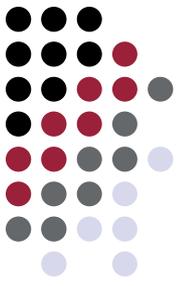
Prof. Dr. George H. G. Fonseca  
Universidade Federal de Ouro Preto



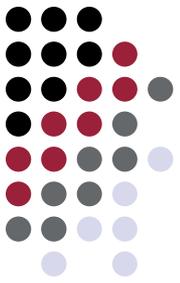


- Ótimo local x Ótimo global

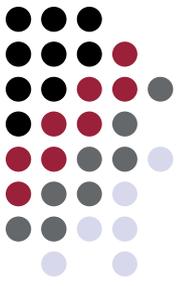




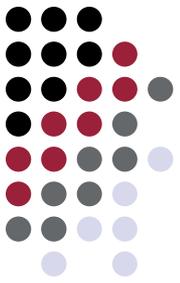
- Procedimentos destinados a encontrar uma boa solução, eventualmente a ótima
  - Aplica, em cada passo, uma heurística subordinada
- Ao contrário das heurísticas convencionais, as metaheurísticas são de **carater geral**
- Providas de mecanismos para **escapar de ótimos locais**
- Diferenciam-se pelo mecanismo para escapar dos ótimos locais



- Duas categorias:
  - Busca local
    - A exploração do espaço de busca é feita por meio de movimentos aplicados, a cada passo, sobre a solução corrente, gerando outra solução promissora em sua vizinhança
  - Principais métodos
    - Multi-start
    - Busca Tabu
    - Simulated Annealing
    - GRASP
    - Variable Neighborhood Search
    - Iterated Local Search

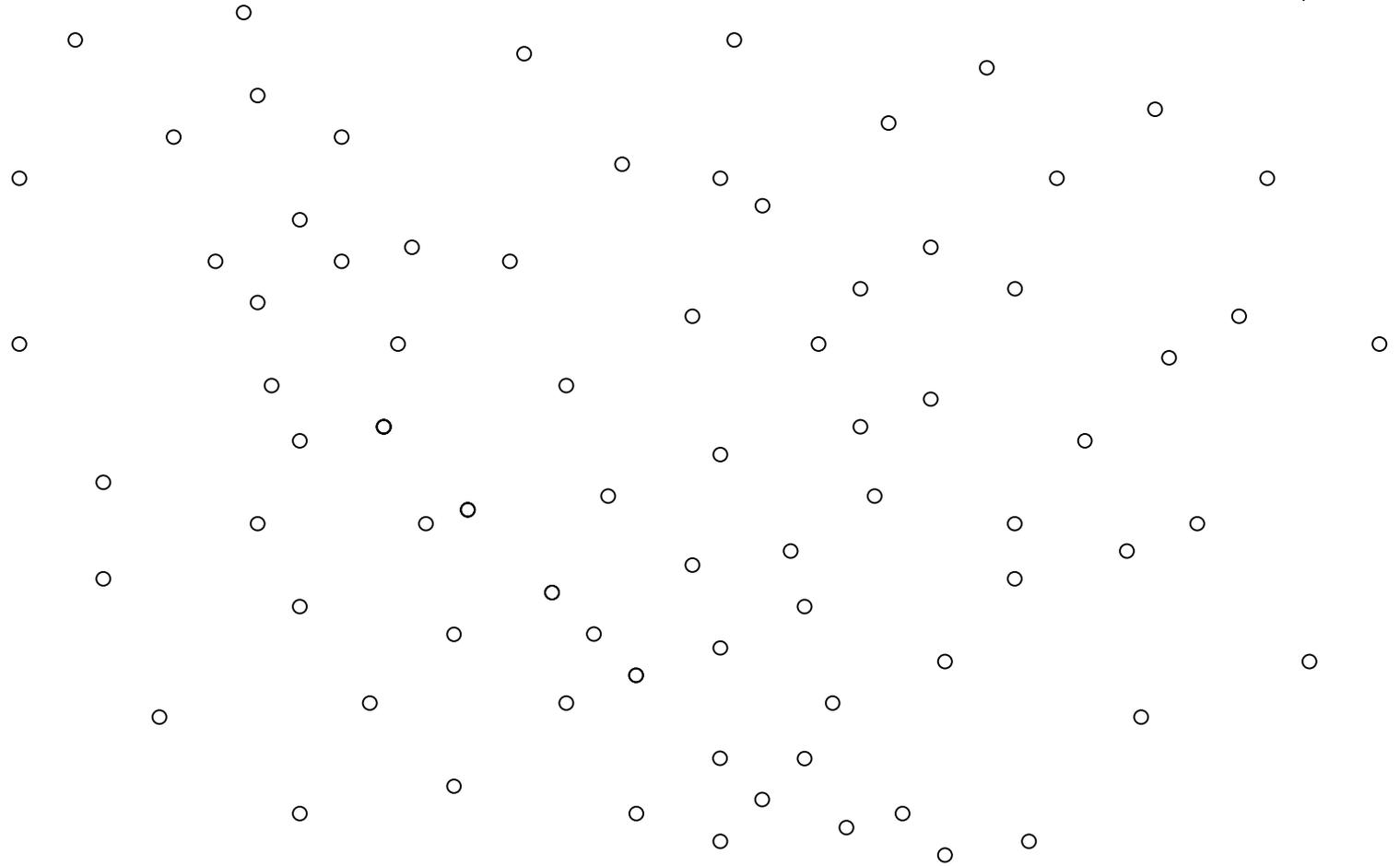
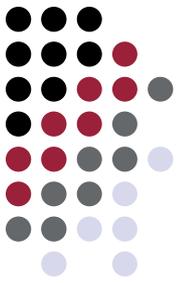


- Duas categorias:
  - Busca populacional
    - Mantém um conjunto de boas soluções (população) e as combina de forma a tentar produzir soluções ainda melhores
  - Principais métodos
    - Algoritmos Genéticos
    - Estratégias Evolutivas
    - Particle Swarm Optimization
    - Colônia de Formigas

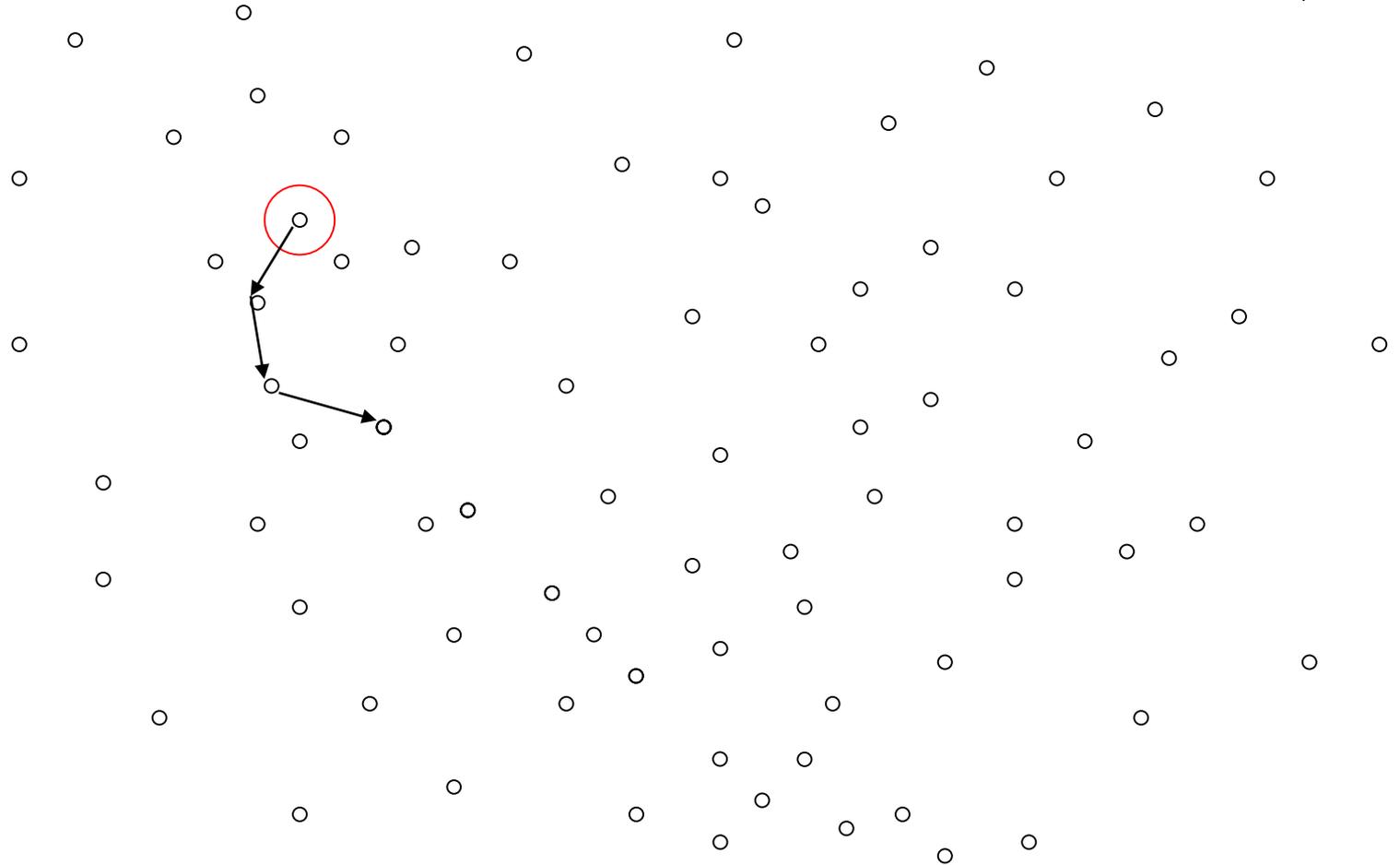
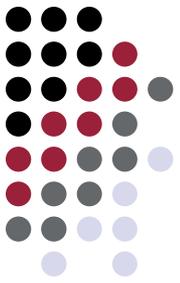


- Faz amostragem no espaço de soluções, aplicando a cada solução inicial aleatória um procedimento de refinamento
- Diversificação do espaço de busca, permitindo escapar de ótimos locais
- Para quando um critério de parada é atingido (tempo ou número de iterações)
- Fácil implementação

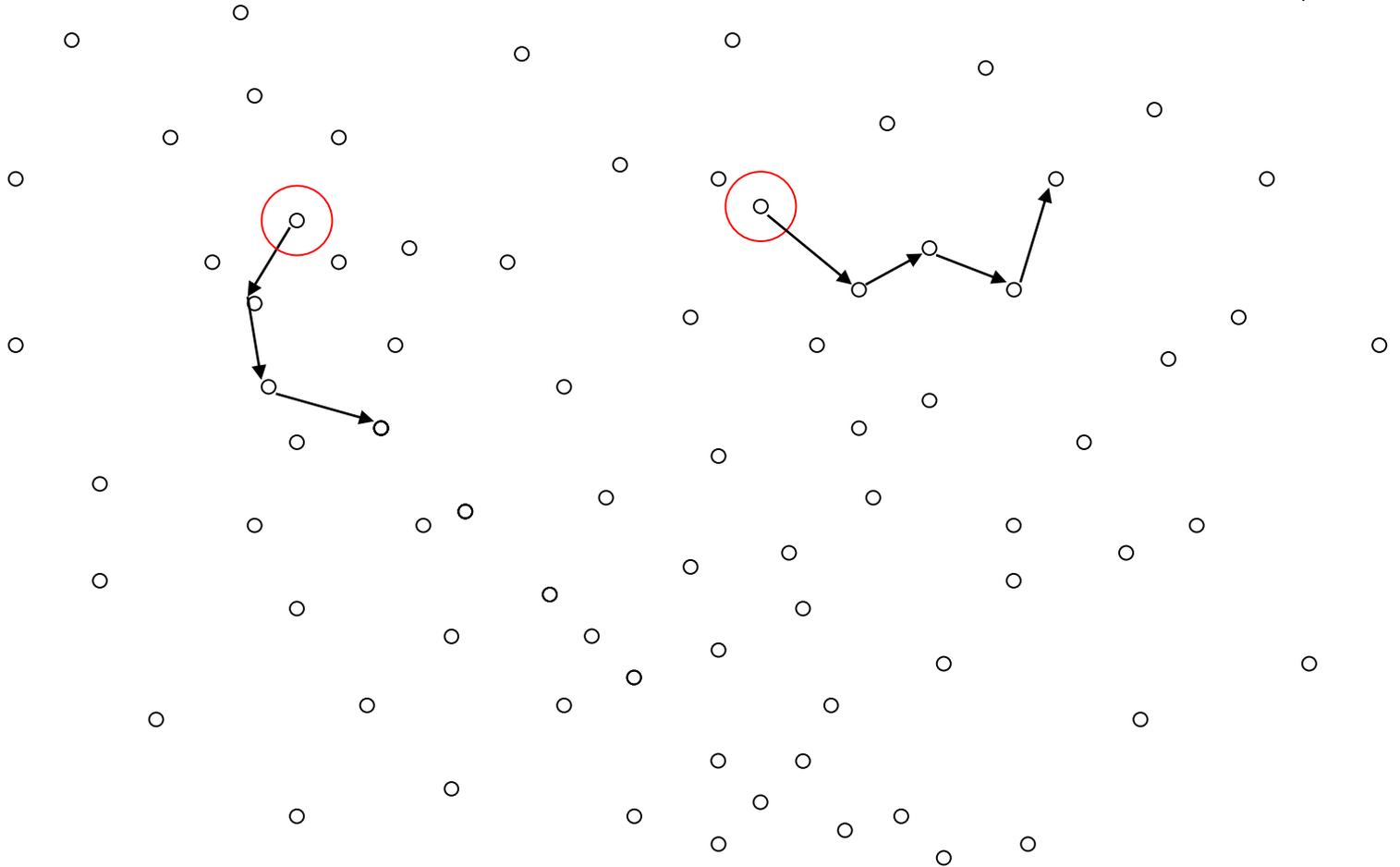
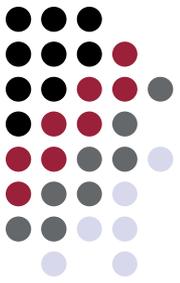
# Multi-start



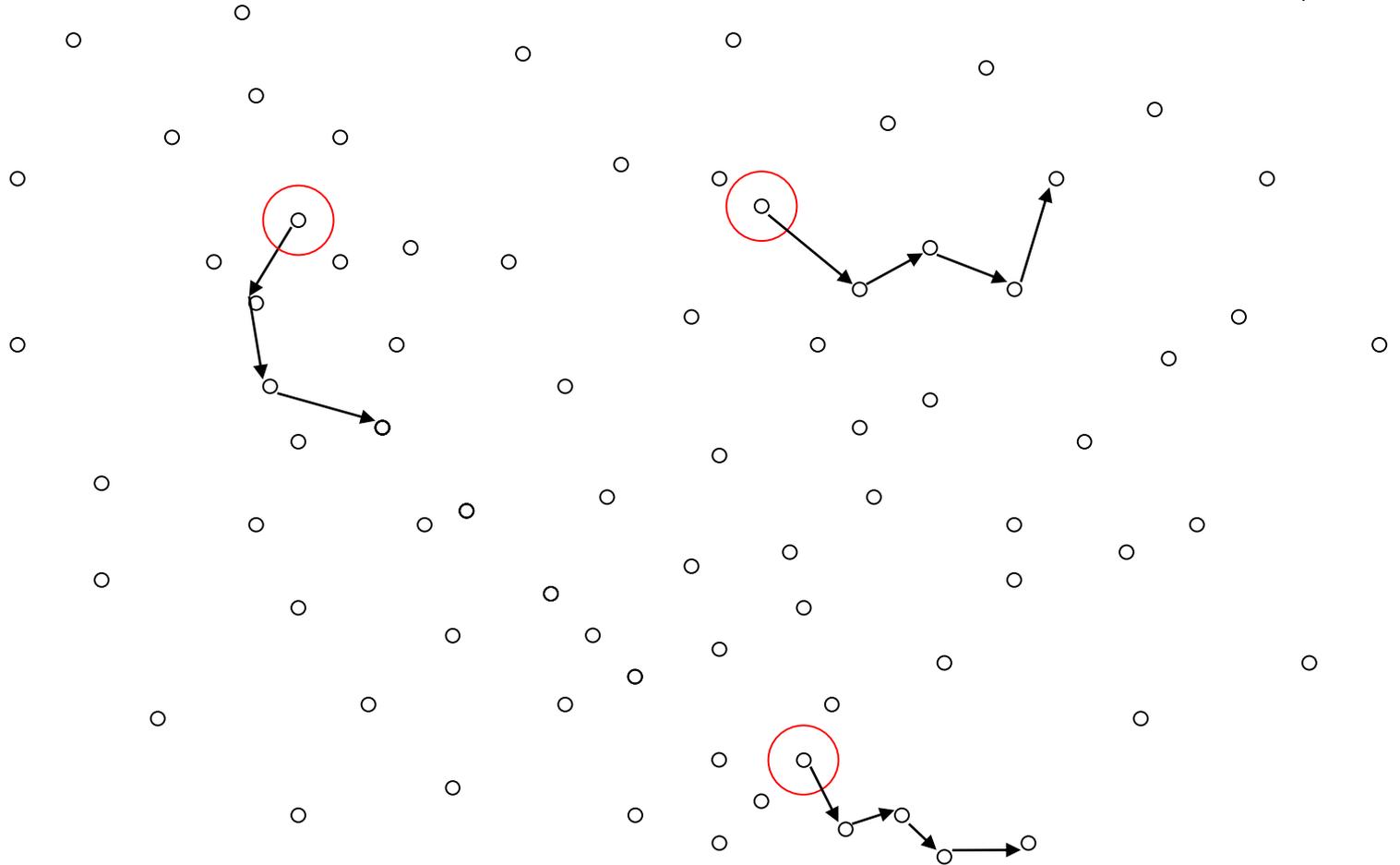
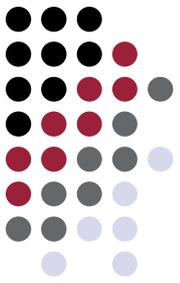
# Multi-start

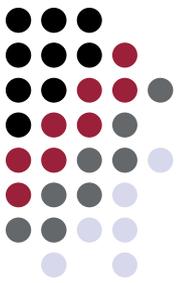


# Multi-start



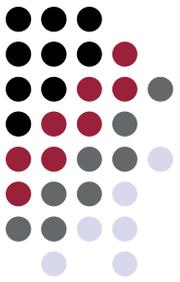
# Multi-start



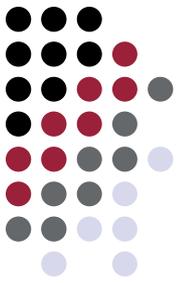


**procedimento** *MultiStart*( $f(\cdot)$ ,  $N(\cdot)$ , *CriterioParada*,  $s$ )

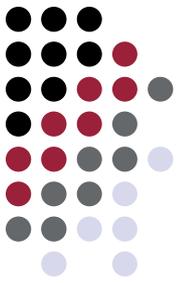
```
1   $f^* \leftarrow \infty$ ;           {Valor associado a  $s^*$  }
2  enquanto (Critério de parada não atendido) faça
3       $s \leftarrow ConstruaSolucao()$ ;  {Gere uma solução  $s$  do espaço de soluções}
4       $s \leftarrow BuscaLocal(s)$ ;      {Aplique um procedimento de melhora em  $s$ }
5      se ( $f(s) < f(s^*)$ ) então
6           $s^* \leftarrow s$ ;
7           $f^* \leftarrow f(s)$ ;
8      fim-se;
9  fim-enquanto;
10  $s \leftarrow s^*$ ;
11 Retorne  $s$ ;
fim MultiStart;
```



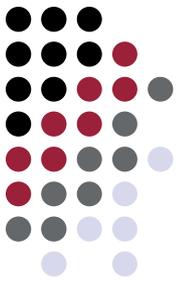
- Proposto por Feo e Resende (1995)
- GRASP - Greedy Randomized Adaptive Search Procedure
- Método iterativo que consiste de duas fases
  - Construção: na qual uma solução é gerada
  - Busca Local: na qual um ótimo local na vizinhança da solução construída é pesquisado
- A melhor solução ao longo de todas as iterações é retornada como resultado



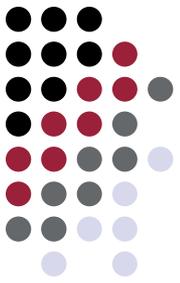
- Na fase de construção um parâmetro  $\alpha = [0.0, 1.0]$  define o quão gulosa/aleatória a solução será
  - $\alpha = 1$  indica uma solução totalmente gulosa enquanto que  $\alpha = 0$  indica uma solução totalmente aleatória
- A eficiência da fase da busca local depende da qualidade da solução inicial construída



- Trade-off sobre  $\alpha$ 
  - Valores altos de  $\alpha$  implicam em soluções iniciais de **boa qualidade**, acelerando o processo de busca local, porém provocam uma baixa diversidade de soluções construídas
  - Valores baixos de  $\alpha$  levam a uma grande **diversidade de soluções**, mas essas soluções são de baixa qualidade, tornando mais lento o processo de busca local
- É possível introduzir estratégias adaptativas para o parâmetro
  - Ajustar de acordo com os resultados obtidos nas iterações anteriores (PRAIS e RIBEIRO, 1999)



```
procedimento  $GRASP(f(\cdot), g(\cdot), N(\cdot), GRASP_{max}, s)$   
1  $f^* \leftarrow \infty$ ;  
2 para ( $Iter = 1, 2, \dots, GRASP_{max}$ ) faça  
3    $Construcao(g(\cdot), \alpha, s)$ ;  
4    $BuscaLocal(f(\cdot), N(\cdot), s)$ ;  
5   se ( $f(s) < f^*$ ) então  
6      $s^* \leftarrow s$ ;  
7      $f^* \leftarrow f(s)$ ;  
8   fim-se;  
9 fim-para;  
10  $s \leftarrow s^*$ ;  
11 Retorne  $s$ ;  
fim  $GRASP$ 
```



- Souza, M. J. F. **Inteligência Computacional para Otimização**. Disponível em [www.iceb.ufop.br/decom/prof/marcone/ico2009](http://www.iceb.ufop.br/decom/prof/marcone/ico2009), acessado em Agosto de 2019.
- Feo, T. A. and Resende, M. G. C. **Greedy randomized adaptive search procedures**. *Journal of Global Optimization*, 6:109-133, 1995.